

Date
02/08/21

Nearaj

Appari

Appari

SDS

Page No.

Date

A. I

Practical - 04

Aim: Implement decision tree learning algorithm for the restaurant seating problem
Implement decision tree learning algorithm for any secondary dataset

Wkups

- 1) Decision tree based learning is classification based learning for discrete values and regression for continuous values
- 2) Set of values are given input to decision tree and it gives output based on classification & prediction
- 3) Decision Tree represents protocols which can be easily understood by humans. It is a classifier which can be represented in the tree structure where each node is leaf node or decision node
- 4) The characteristic of decision tree are
 - a) Root node is starting node
 - b) Decision node gives the test to be carried out or decision
 - c) Leaf node stand for probable final decision
 - d) Every node in the tree returns yes/no/ probable decision
 - e) Branches of tree are labeled with probable vectors

Algorithms

- 1) Start
- 2) Import modules (panda, sklearn.tree, pydotplus, DecisionTreeClassifier, matplotlib.pyplot, matplotlib.image, sklearn, sklearn.tree)
- 3) Read csv file
- 4) Converting the values to float of csv file
- 5) Define X and y
- 6) Use graphs to create graph
- 7) Show graph in png file
- 8) Predict tree

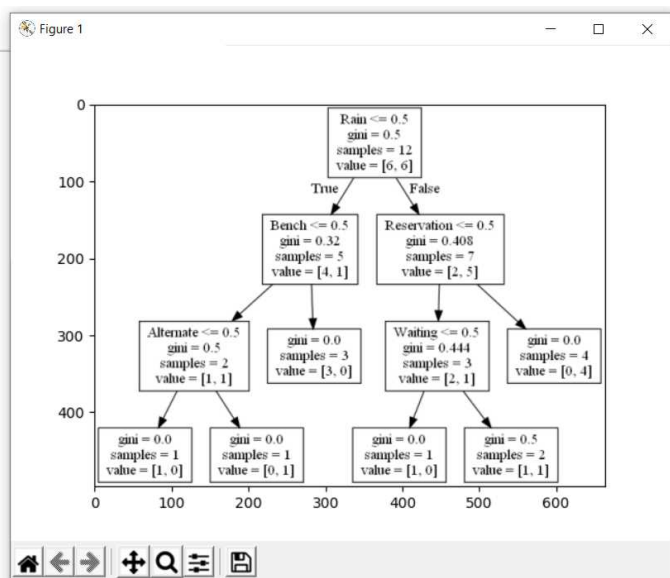
p4 try 2.py - E:\fffiles\college pracs and projects\AI\p4 try 2.py (3.8.3)
File Edit Format Run Options Window Help

```
import pandas
from sklearn import tree
import pydotplus
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
import matplotlib.image as pltimg
#Neeraj Appari
```

```
df = pandas.read_csv("restuarant2.csv")
d = {'Yes': 1, 'No': 0}
df['Alternate'] = df['Alternate'].map(d)
d = {'Yes': 1, 'No': 0}
df['Bench'] = df['Bench'].map(d)
d = {'Yes': 1, 'No': 0}
df['Hungry'] = df['Hungry'].map(d)
d = {'Yes': 1, 'No': 0}
df['Waiting'] = df['Waiting'].map(d)
d = {'Yes': 1, 'No': 0}
df['Costly'] = df['Costly'].map(d)
d = {'Yes': 1, 'No': 0}
df['Rain'] = df['Rain'].map(d)
d = {'Yes': 1, 'No': 0}
df['Reservation'] = df['Reservation'].map(d)
d = {'Yes': 1, 'No': 0}
df['Patrons'] = df['Patrons'].map(d)
d = {'Yes': 1, 'No': 0}
```

```
features = [ 'Alternate', 'Bench', 'Hungry', 'Waiting', 'Costly', 'Rain', 'Reservation', 'Patrons']
```

```
X = df[features]
y = df['Wait']
```



p4 try 2.py - E:\fffiiles\college pracs and projects\AI\p4 try 2.py (3.8.3)

File Edit Format Run Options Window Help

```
df['Hungry'] = df['Hungry'].map(d)
d = {'Yes': 1, 'No': 0}
df['Waiting'] = df['Waiting'].map(d)
d = {'Yes': 1, 'No': 0}
df['Costly'] = df['Costly'].map(d)
d = {'Yes': 1, 'No': 0}
df['Rain'] = df['Rain'].map(d)
d = {'Yes': 1, 'No': 0}
df['Reservation'] = df['Reservation'].map(d)
d = {'Yes': 1, 'No': 0}
df['Patrons'] = df['Patrons'].map(d)
d = {'Yes': 1, 'No': 0}

features = [ 'Alternate', 'Bench', 'Hungry', 'Waiting', 'Costly', 'Rain', 'Reservation', 'Patrons' ]

X = df[features]
y = df['Wait']

dtree = DecisionTreeClassifier()
dtree = dtree.fit(X, y)
data = tree.export_graphviz(dtree, out_file=None, feature_names=features)
graph = pydotplus.graph_from_dot_data(data)
graph.write_png('mydecisiontree.png')

img=pltimg.imread('mydecisiontree.png')
imgplot = plt.imshow(img)
plt.show()

print(dtree.predict([[1, 0, 1, 1, 0, 1, 1, 1]]))
```

Python 3.8.3 Shell

File Edit Shell Debug Options Window Help

Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020) [AMD64] on win32

Type "help", "copyright", "credits" or "license()" for more

>>>

===== RESTART: E:\fffiiles\college pracs and projects\AI\p4 try 2.py

py =====

['Yes']

>>> |

Type here to search



09:30
02-08-2021

restuarant2 - Excel

neeraj appari

FileHomeInsertPage LayoutFormulasDataReviewViewHelpTell me what you want to do

ClipboardFontAlignmentNumberStylesCellsEditing

Conditional FormattingFormat as TableCell StylesInsertDeleteFormat

AutoSumFillSort & FilterFind & Select

Document Recovery

Excel has recovered the following files. Save the ones you wish to keep.

Literature Review Table Neer...
Version created from the last Au...
01-08-2021 20:46

Literature Review Table Neer...
Version created last time the use...
01-08-2021 01:02

Which file do I want to save?

Close

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
|----|-----------|-------|--------|---------|--------|------|------------|---------|------|---|---|---|---|---|---|---|---|---|---|
| 1 | Alternate | Bench | Hungry | Waiting | Costly | Rain | Reservatic | Patrons | Wait | | | | | | | | | | |
| 2 | Yes | No | Yes | No | Yes | Yes | Yes | No | Yes | | | | | | | | | | |
| 3 | Yes | Yes | No | No | Yes | No | No | Yes | No | | | | | | | | | | |
| 4 | No | No | Yes | No | Yes | Yes | Yes | No | Yes | | | | | | | | | | |
| 5 | No | Yes | Yes | No | No | No | No | No | No | | | | | | | | | | |
| 6 | Yes | No | Yes | Yes | No | No | No | No | Yes | | | | | | | | | | |
| 7 | No | Yes | Yes | No | No | No | Yes | No | No | | | | | | | | | | |
| 8 | No | Yes | Yes | No | Yes | Yes | Yes | No | Yes | | | | | | | | | | |
| 9 | No | No | No | No | Yes | Yes | No | Yes | No | | | | | | | | | | |
| 10 | Yes | Yes | No | No | No | Yes | Yes | Yes | Yes | | | | | | | | | | |
| 11 | Yes | No | No | Yes | No | Yes | No | Yes | No | | | | | | | | | | |
| 12 | Yes | No | No | Yes | No | Yes | No | Yes | Yes | | | | | | | | | | |
| 13 | No | No | Yes | Yes | Yes | No | Yes | No | No | | | | | | | | | | |
| 14 | | | | | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | | | | | |
| 16 | | | | | | | | | | | | | | | | | | | |
| 17 | | | | | | | | | | | | | | | | | | | |
| 18 | | | | | | | | | | | | | | | | | | | |
| 19 | | | | | | | | | | | | | | | | | | | |
| 20 | | | | | | | | | | | | | | | | | | | |
| 21 | | | | | | | | | | | | | | | | | | | |
| 22 | | | | | | | | | | | | | | | | | | | |
| 23 | | | | | | | | | | | | | | | | | | | |
| 24 | | | | | | | | | | | | | | | | | | | |
| 25 | | | | | | | | | | | | | | | | | | | |
| 26 | | | | | | | | | | | | | | | | | | | |
| 27 | | | | | | | | | | | | | | | | | | | |
| 28 | | | | | | | | | | | | | | | | | | | |
| 29 | | | | | | | | | | | | | | | | | | | |

Ready

Type here to search

Google Chrome

File Explorer

Microsoft Edge

Visual Studio Code

PowerShell

SQL Server Enterprise

Excel

Task Manager

System Restore

Windows Defender

Windows Firewall

Windows Update

Windows Defender Security Center

Windows Defender Firewall

Windows Defender SmartScreen

Windows Defender Security Center

Windows Defender Firewall

Windows Defender SmartScreen

09:30

02-08-2021

ENG

p4 try 2.py - E:\fffiles\college pracs and projects\AI\p4 try 2.py (3.8.3)

File Edit Format Run Options Window Help

```
df['Hungry'] = df['Hungry'].map(d)
d = {'Yes': 1, 'No': 0}
df['Waiting'] = df['Waiting'].map(d)
d = {'Yes': 1, 'No': 0}
df['Costly'] = df['Costly'].map(d)
d = {'Yes': 1, 'No': 0}
df['Rain'] = df['Rain'].map(d)
d = {'Yes': 1, 'No': 0}
df['Reservation'] = df['Reservation'].map(d)
d = {'Yes': 1, 'No': 0}
df['Patrons'] = df['Patrons'].map(d)
d = {'Yes': 1, 'No': 0}

features = [ 'Alternate', 'Bench', 'Hungry', 'Waiting', 'Costly', 'Rain', 'Reservation', 'Patrons' ]

X = df[features]
y = df['Wait']

dtree = DecisionTreeClassifier()
dtree = dtree.fit(X, y)
data = tree.export_graphviz(dtree, out_file=None, feature_names=features)
graph = pydotplus.graph_from_dot_data(data)
graph.write_png('mydecisiontree.png')

img=pltimg.imread('mydecisiontree.png')
imgplot = plt.imshow(img)
plt.show()

print(dtree.predict([[1, 0, 1, 0, 0, 1, 0, 0]]))
```

Ln: 46 Col: 0

Type here to search

09:31 02-08-2021 ENG

p4 try 2.py - E:\fffiles\college pracs and projects\AI\p4 try 2.py (3.8.3)

File Edit Format Run Options Window Help

```
df['Hungry'] = df['Hungry'].map(d)
d = {'Yes': 1, 'No': 0}
df['Waiting'] = df['Waiting'].map(d)
d = {'Yes': 1, 'No': 0}
df['Costly'] = df['Costly'].map(d)
d = {'Yes': 1, 'No': 0}
df['Rain'] = df['Rain'].map(d)
d = {'Yes': 1, 'No': 0}
df['Reservation'] = df['Reservation'].map(d)
d = {'Yes': 1, 'No': 0}
df['Patrons'] = df['Patrons'].map(d)
d = {'Yes': 1, 'No': 0}

features = [ 'Alternate', 'Bench', 'Hungry', 'Waiting', 'Costly', 'Rain', 'Reservation', 'Patrons' ]

X = df[features]
y = df['Wait']

dtree = DecisionTreeClassifier()
dtree = dtree.fit(X, y)
data = tree.export_graphviz(dtree, out_file=None, feature_names=features)
graph = pydotplus.graph_from_dot_data(data)
graph.write_png('mydecisiontree.png')

img=pltimg.imread('mydecisiontree.png')
imgplot = plt.imshow(img)
plt.show()

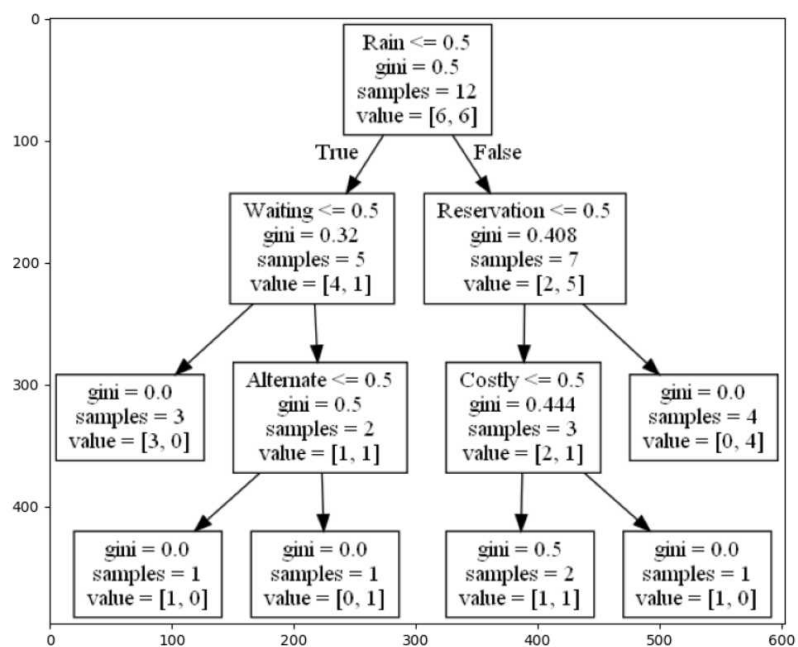
print(dtree.predict([[1, 0, 1, 0, 0, 1, 0, 0]]))
```

Ln: 46 Col: 0

Type here to search

09:31 02-08-2021 ENG

Figure 1




```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\fffiiles\college pracs and projects\AI\p4 try 2.py =====
['No']
>>>
```

p4 try 2-b.py - E:\ffiles\college pracs and projects\AI\p4 try 2-b.py (3.8.3)

File Edit Format Run Options Window Help

```
import pandas
from sklearn import tree
import pydotplus
from sklearn.tree import DecisionTreeClassifier
import matplotlib.pyplot as plt
import matplotlib.image as pltimg
#Neeraj Appari|

df = pandas.read_csv("play.csv")

d = {'Yes': 1, 'No': 0}
df['Ground'] = df['Ground'].map(d)
d = {'Yes': 1, 'No': 0}
df['Day'] = df['Day'].map(d)
d = {'Yes': 1, 'No': 0}
df['Weekend'] = df['Weekend'].map(d)
d = {'Yes': 1, 'No': 0}
df['Costly'] = df['Costly'].map(d)
d = {'Yes': 1, 'No': 0}
df['Rain'] = df['Rain'].map(d)
d = {'Yes': 1, 'No': 0}
df['Team'] = df['Team'].map(d)
d = {'Yes': 1, 'No': 0}

features = [ 'Ground', 'Day', 'Weekend', 'Costly', 'Rain', 'Team']

X = df[features]
y = df['Play']

dtree = DecisionTreeClassifier()
dtree = dtree.fit(X, y)
data = tree.export_graphviz(dtree, out_file=None, feature_names=features)
```

Ln: 7 Col: 14

Type here to search



09:54
02-08-2021

p4 try 2-b.py - E:\fffiles\college pracs and projects\AI\p4 try 2-b.py (3.8.3)

File Edit Format Run Options Window Help

```
df['Ground'] = df['Ground'].map(d)
d = {'Yes': 1, 'No': 0}
df['Day'] = df['Day'].map(d)
d = {'Yes': 1, 'No': 0}
df['Weekend'] = df['Weekend'].map(d)
d = {'Yes': 1, 'No': 0}
df['Costly'] = df['Costly'].map(d)
d = {'Yes': 1, 'No': 0}
df['Rain'] = df['Rain'].map(d)
d = {'Yes': 1, 'No': 0}
df['Team'] = df['Team'].map(d)
d = {'Yes': 1, 'No': 0}

features = [ 'Ground', 'Day', 'Weekend', 'Costly', 'Rain', 'Team']

X = df[features]
y = df['Play']

dtree = DecisionTreeClassifier()
dtree = dtree.fit(X, y)
data = tree.export_graphviz(dtree, out_file=None, feature_names=features)
graph = pydotplus.graph_from_dot_data(data)
graph.write_png('mydecisiontree.png')

img=pltimg.imread('mydecisiontree.png')
imgplot = plt.imshow(img)
plt.show()

print(dtree.predict([[1, 0, 1, 0, 0, 0]]))
```

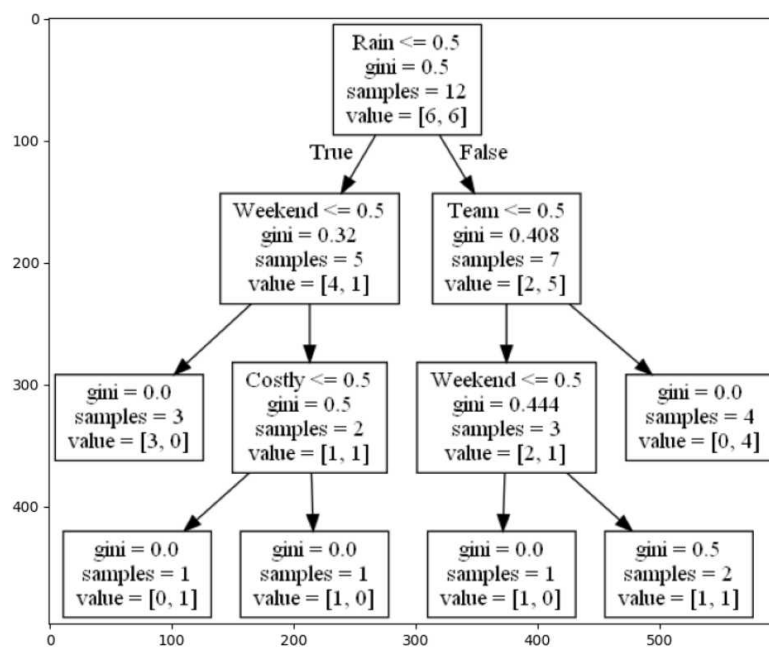
Ln: 7 Col: 14

Type here to search



ENG 09:54 02-08-2021

Figure 1



```
p4 try 2-b.py Python 3.8.3 Shell
File Edit For File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:/fffiiles/college pracs and projects/AI/p4 try 2-b.py =====
>>> ['Yes']
>>> |
df['Ra
d = {'
df['Te
d = {'
featur
X = df
y = df
dtree
dtree
data =
graph
graph.
img=pl
imgplo
plt.sh
print(
```

Windows taskbar showing search bar, microphone, and various application icons (Chrome, Edge, File Explorer, etc.). System clock shows 09:55 on 02-08-2021.